

The background is a dark chalkboard with various white chalk sketches. In the top left, there's a large 'V' and a globe. Below the globe is a microscope. In the bottom left, there's a stack of books. In the bottom center, there's an open book with some handwritten text. In the bottom right, there are mathematical symbols like a percentage sign, a plus sign, and a less-than sign.

## **Anaka :**

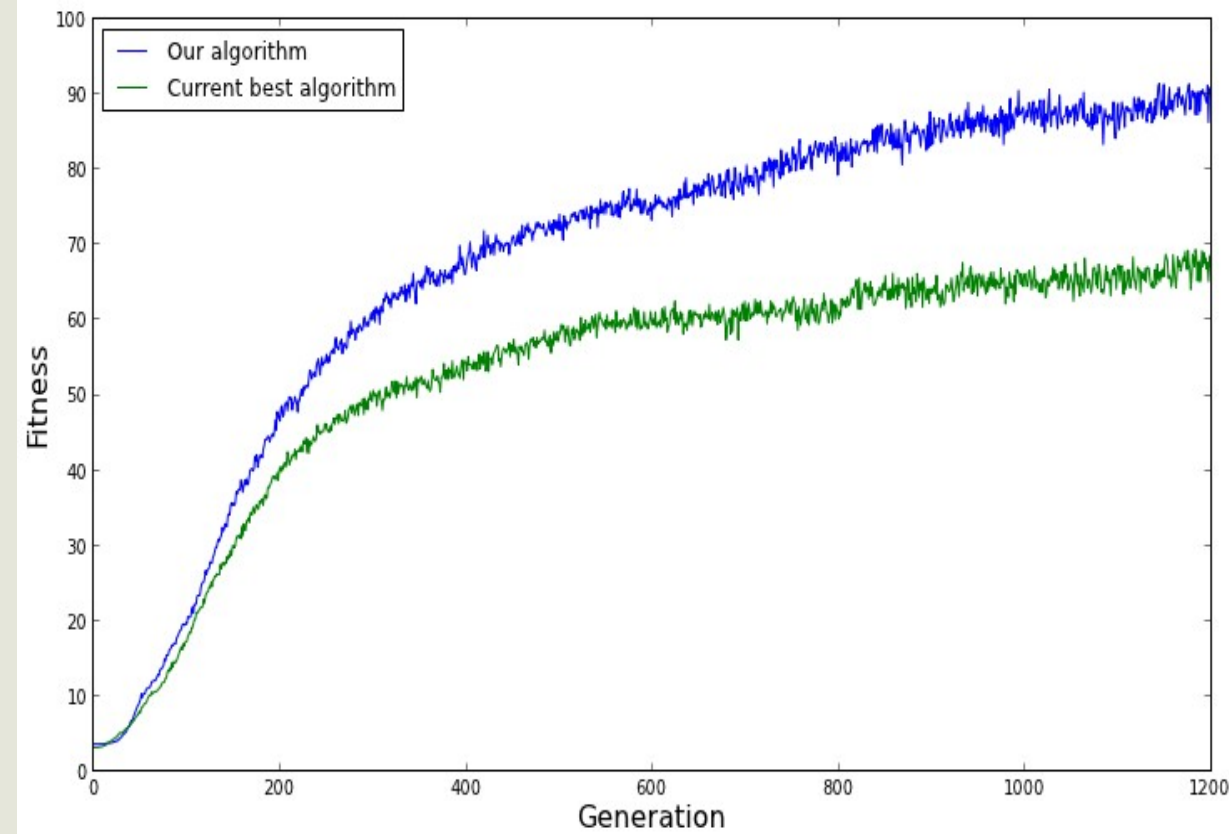
Efficient Prediction of Time-Based Events using Parallel Computing Architecture.

**Team :** Ankit Sirmorya, Preethu Thomas, Shantanu Kande, Sumeet Pande

# Project Objectives



- Primary objective of the project is reliable, quick and efficient prediction of time based events by making use of parallel computing paradigm.
- A hybrid model based on Genetic Algorithm(GA) and Fuzzy Time Series(FTS) is being proposed to be implemented on time bound data set.
- GA+FTS to be used for the purpose of distributed processing of time bound data set using MapReduce on Hadoop framework.

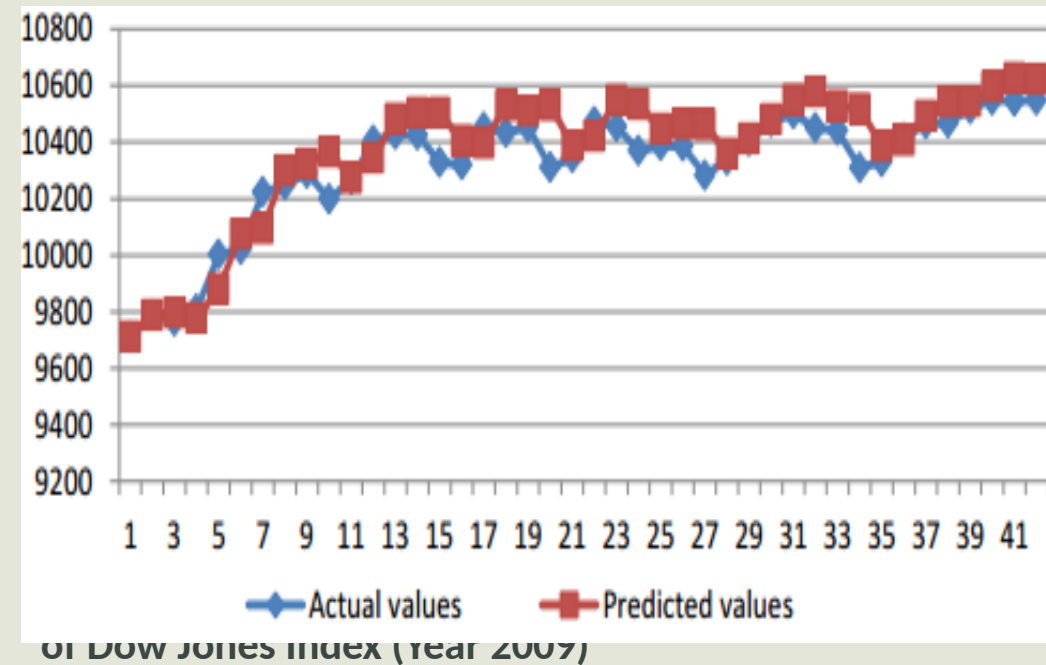


## Related Work

Examples where such methodologies have been implemented successfully:

- Enrollment Forecasting Model.
- Stock Market Prediction Model.
- Temperature Prediction Model.

Following figure shows the results for implementation of Stock Market Index Prediction Model for Dow Jones Index.



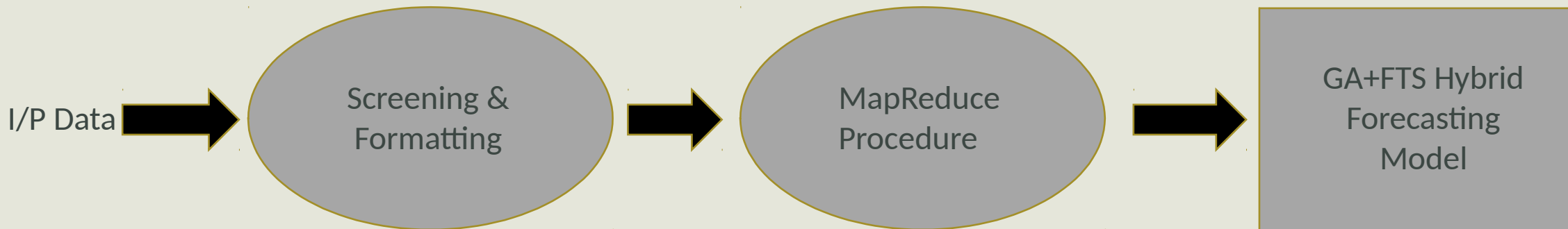


# Problem Formulation

- Statement : Given an input data set consisting of statistical data pertaining to number of events per month for a range of yearly input (say starting from 1971), implement an efficient forecasting model that can predict the number of events for future years.
- Challenges :
  1. Input Data Parsing as per MapReduce Paradigm.
  2. Implementation of GA+FTS prediction model.
  3. Adapting this GA + FTS model so as to implement the same in MapReduce Paradigm.
  4. Explore various Result visualization methods.

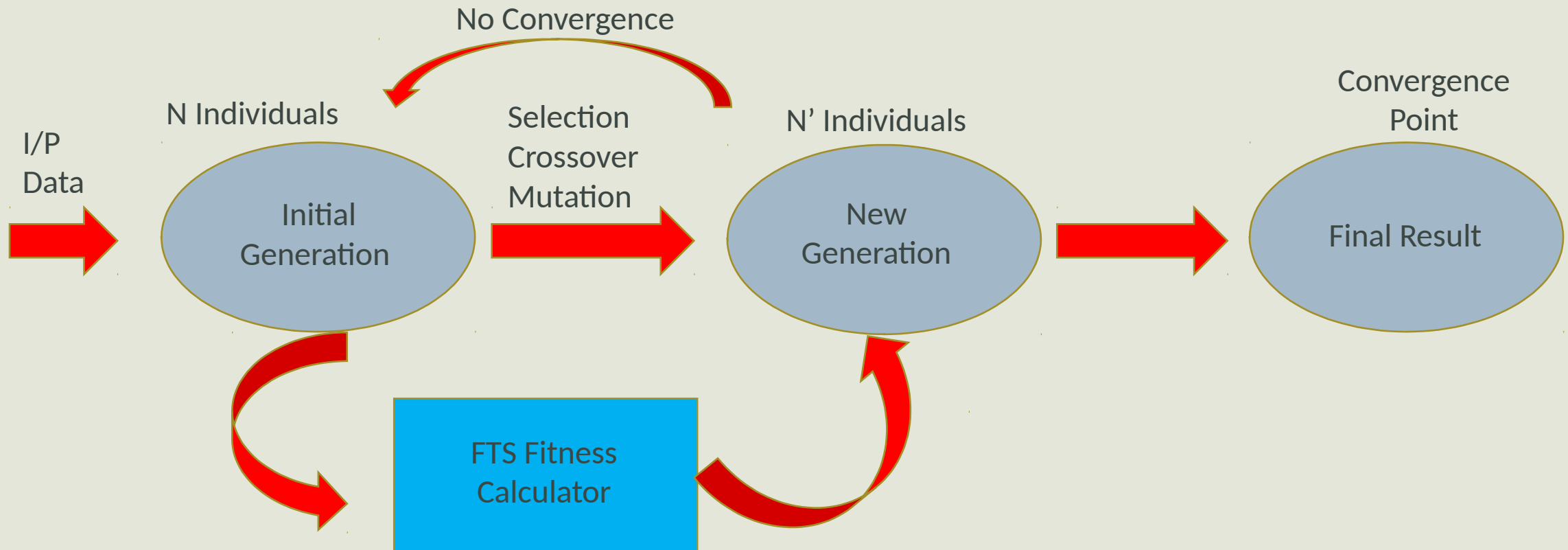
# System Architecture Phase 1 : Data Ingestion

- Main Objective of this phase is to allow a configurable way of parsing the input data. Input data is obtained from statistical data sets available publicly relevant to our model.
- The data needs to be processed by application of MapReduce Paradigm so as to fine tune the data as per the specification of our forecasting model.



## System Architecture Phase 2 : Hybrid GA+FTS Prediction Model

- GA+FTS model can be visualized as a combination of two functionalities where GA is used to improve correctness of each corresponding generation over previous ones while FTS is used as a fitness calculator to check the mean square error (MSE).




## Sub System (Phase 2) : Genetic Algorithm (GA)

- A set of computational methods that are inspired by evolutionary biology such as mutation, selection, inheritance and crossover that can be used to find solutions to optimization problem.


Step 1: Generating random population of possible solution which corresponds to a single generation.



Step 2: Fitness of every solution calculated and multiple solutions selected on basis of their fitness levels to reproduce a new generation.



Step 3: Various GA operators are used such as SELECTION , Crossover and MUTATION to improve the fitness quotient of each individual/generation.



Step 4: Process continues till there is no significant improvement in fitness level of a generation over previous one.

Genetic Algorithm mimics the evolutionary principle “ Survival of the Fittest”



## Sub System (Phase 2) : Fuzzy Time Series (FTS).

Dividing the input data of time bound event into even and equal length of intervals.

Random selection of Chromosome/Individual that is a product of GA iteration.

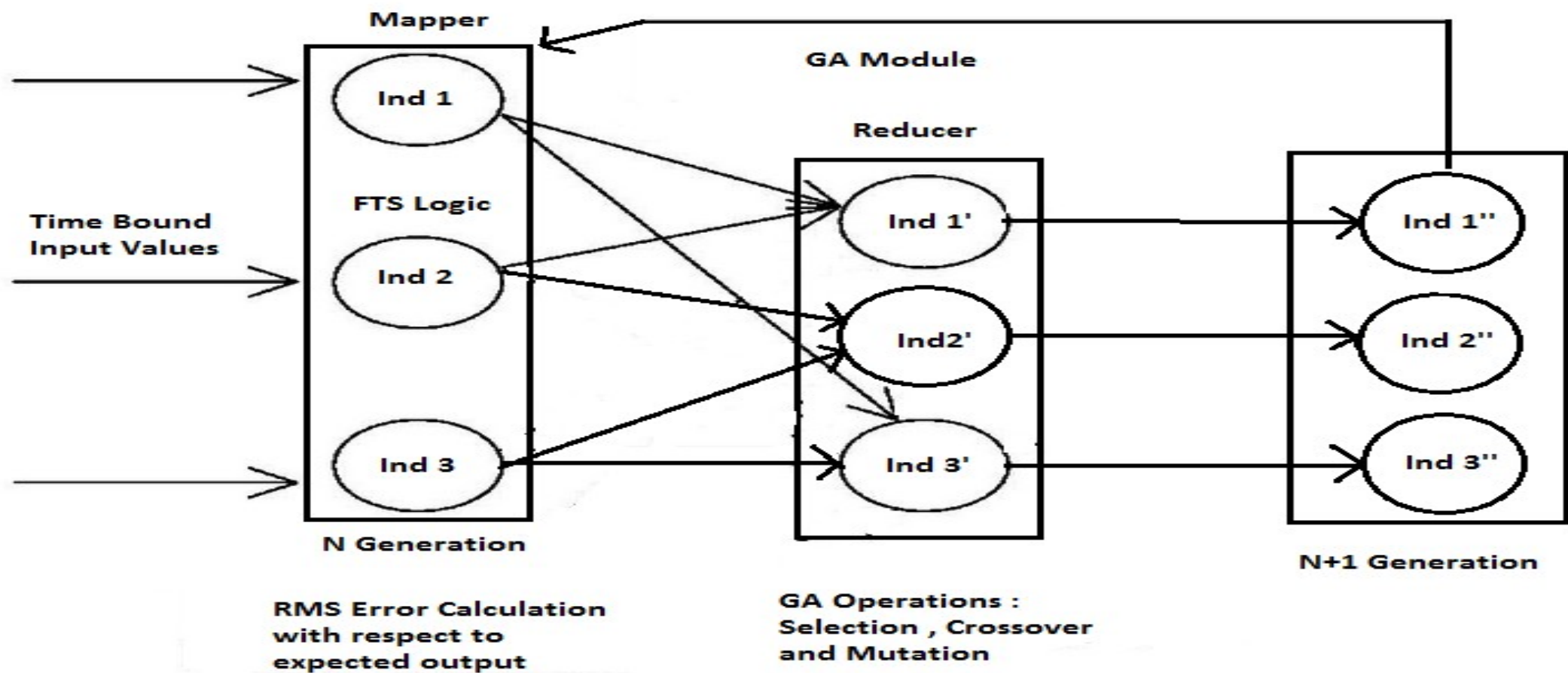
To fuzzify randomly generated set of chromosomes

Decide fuzzy enrollment pattern and also nth order fuzzy logic relationship.

Forecast best results on the basis of Root Mean Square Error (Fitness value parameter).



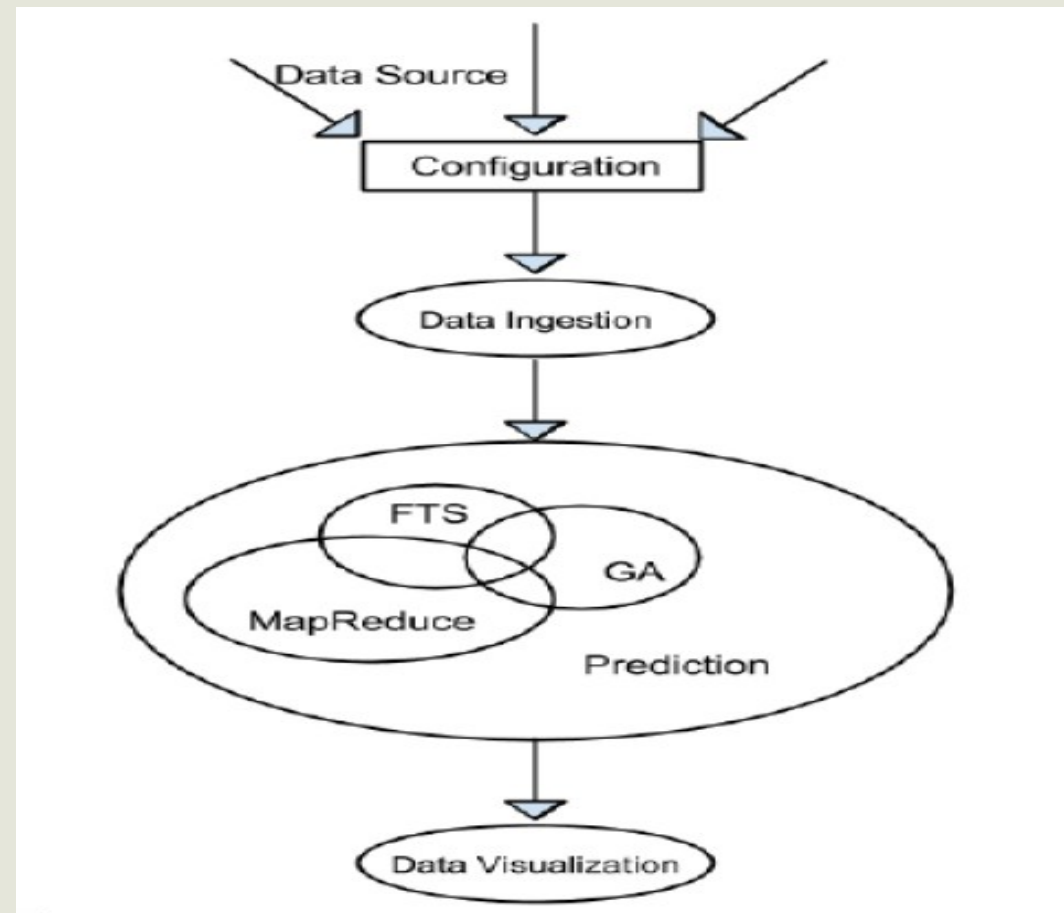
# MapReduce Implementation : GA+FTS Hybrid Model.



## System Architecture Phase 3 : Result Visualization

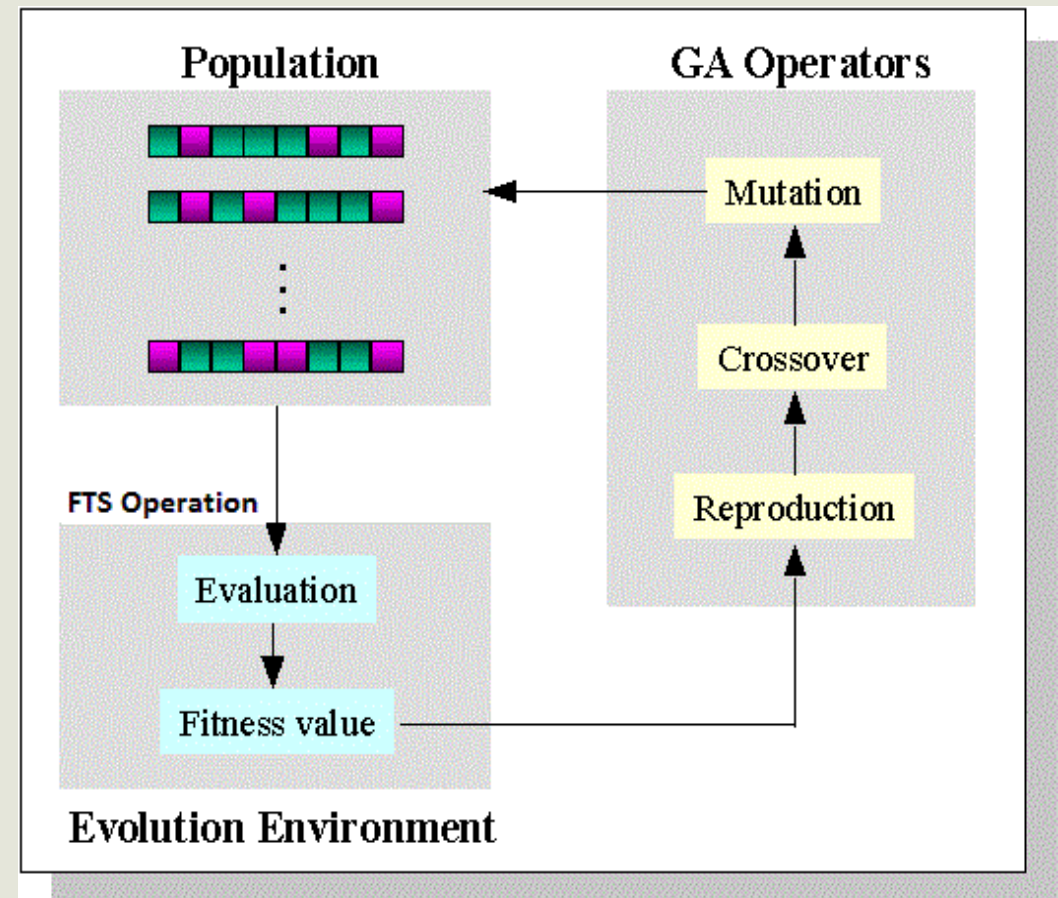
- This phase will make use of prediction model results and display them in comparative graphical format.
- Makes it easy to understand and visualize the results.
- Various tools being explored such as Plotly and charts.js packages.

Fig : System Architecture implementation.



## Block Diagram Operation:

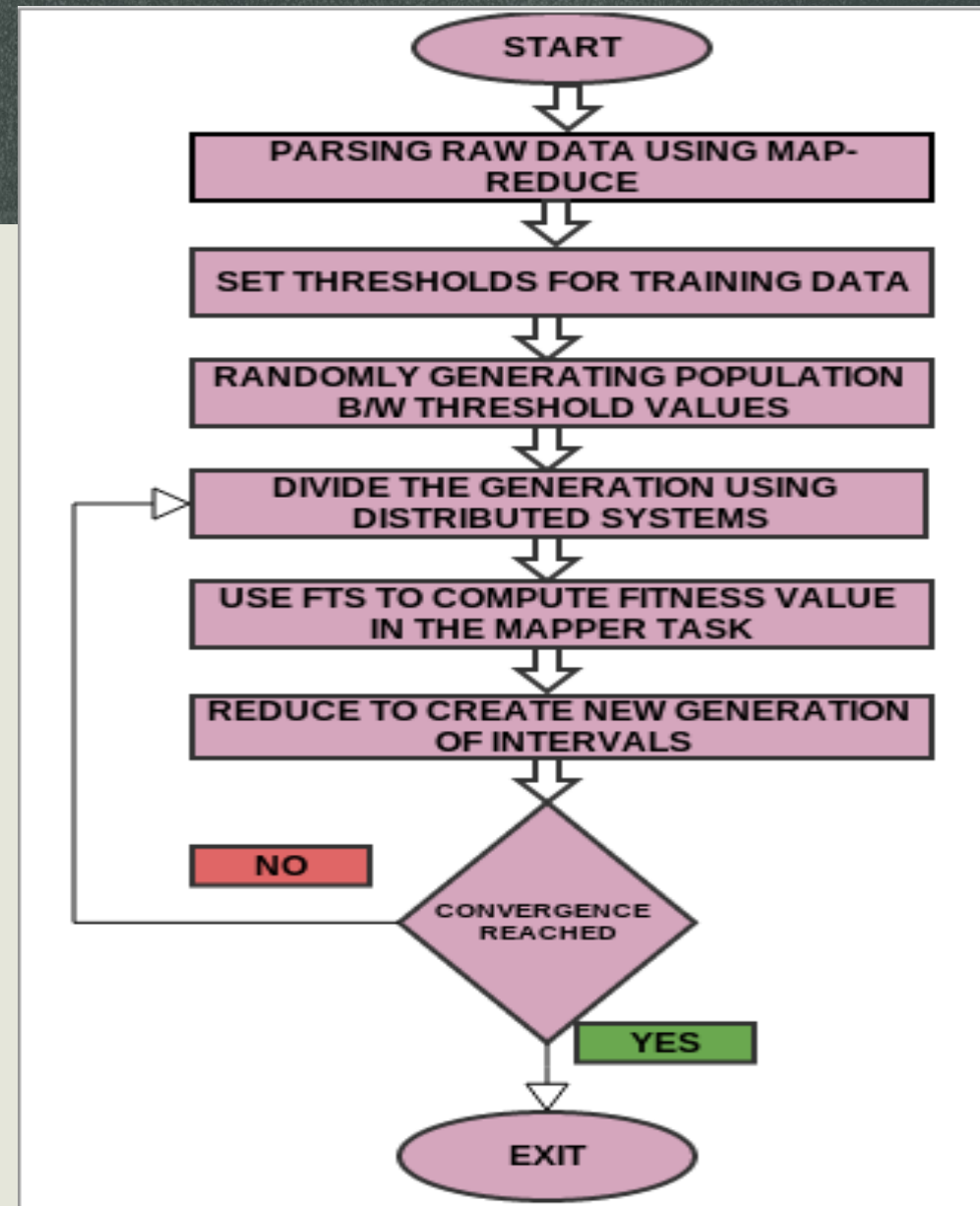
- Basic Idea is to predict from a time bound data set by making use of FTS and GA.
- FTS used as fitness calculator that is to calculate correctness of prediction for each individual as a function of Root Mean Square Error (RMS).
- GA used to improve correctness for each generation over previous generation by making use of GA operators.
- Process will terminate when a convergence point is reached.





# Flowchart Implementation:

- Creation of first generation of Individuals from the time bound data set.
- Dividing this data in distributed manner so as to implement the data in MapReduce framework.
- Mapper Function -> Apply FTS to calculate fitness of each individual.
- Reducer Function -> Perform selection , crossover and mutation (GA operators) thereby creating an a new generation.



# Algorithm for project implementation :

Step 1: After reading the data calculate the maximum and the minimum values and select thresholds so that we get  $L_{min}$  and  $L_{max}$ .

Step 2: Create the first generation of Genetic Algorithm which will be a set of individuals such that each will be a set of buckets which will lie in the range  $L_{min}$  to  $L_{max}$ .

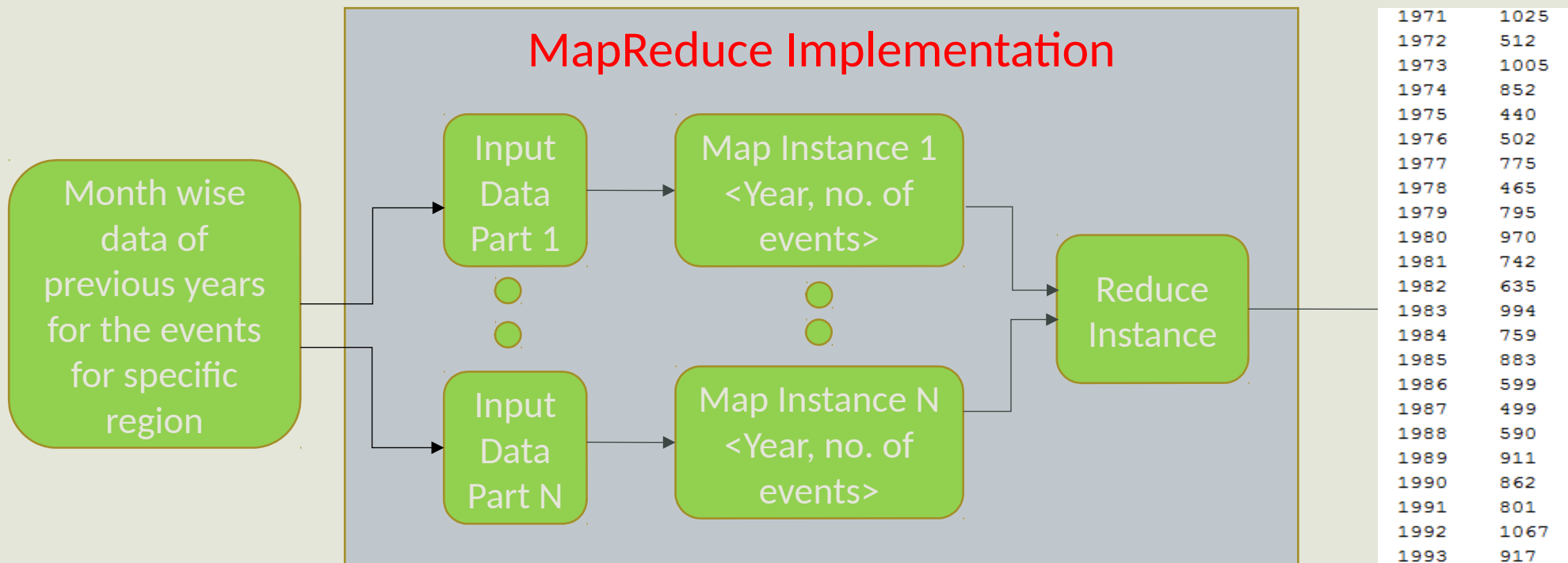
Step 3: Create the division of the generation data which can be used in the MapReduce model. The mapper task will apply FTS and will make prediction for each individual and fitness value for each of them will be calculated.

Step 4: The reducer task will perform the selection, cross-over and mutation operations of GA based on the fitness value calculated by the mapper task. Thus, a new generation of individuals will be created.

Step 5: Repeat Steps 4 through 6 until the convergence point is not reached. Once reached terminate the process.

# Implementation: Data Parsing Module

- Breaks the data into cohesive tokens.
- Interprets the tokens and constructs larger elements based on these tokens.





# Implementation: Prediction Module (GA+FTS) -1

I/P Data  
from the  
parser

1971	1025
1972	512
1973	1005
1974	852
1975	440
1976	502
1977	775
1978	465
1979	795
1980	970
1981	742
1982	635
1983	994
1984	759
1985	883
1986	599
1987	499
1988	590
1989	911
1990	862
1991	801
1992	1067
1993	917

Minimum value =  
440  
Maximum value =  
1067

U [Dmin - D1,  
Dmax- D2]

Selection of  
Universe of  
Discourse

U[400, 1100]  
D1 = 40, D2 = 33

Intervals

A[400, 499]

A[500,599]

A[600,699]

⋮

A[1000,1100]

First Generation

Fuzzify the  
historical data  
& generate  
fuzzy logical  
relationships

Calculate  
fitness value &  
implement GA  
to improve  
fitness

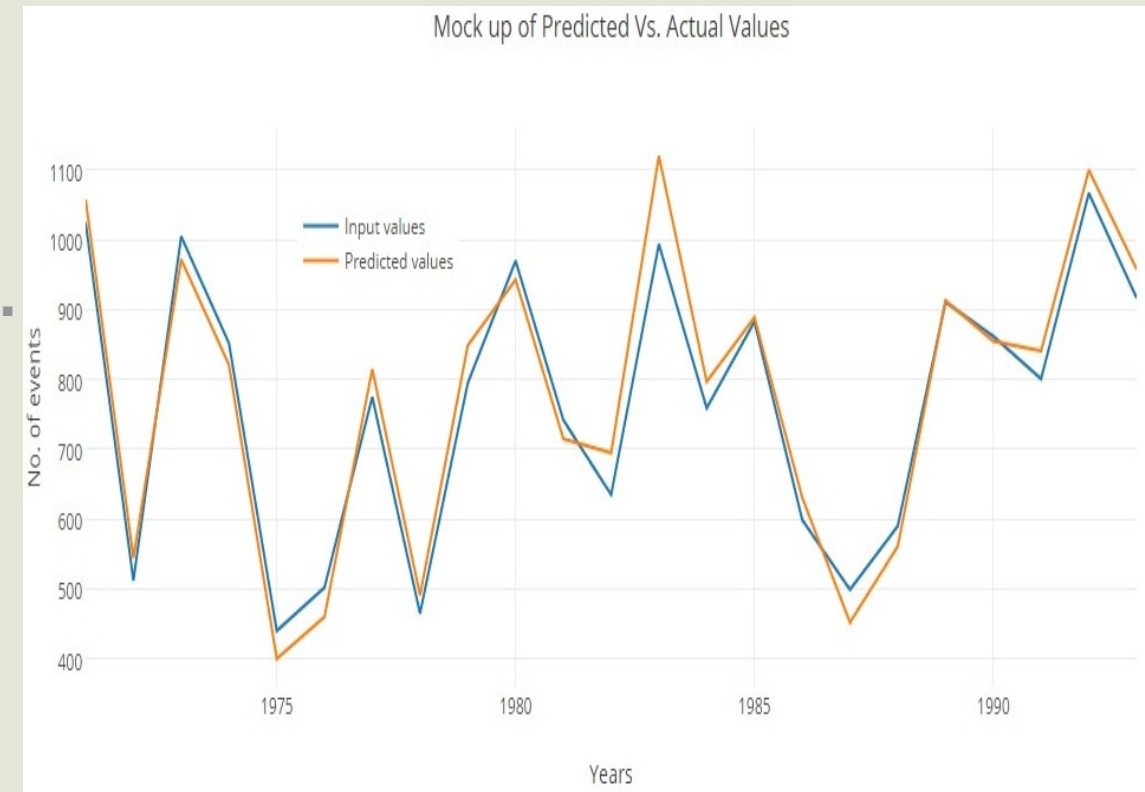
## Implementation: Prediction Module (GA+FTS) -2

- The iteration cycle of GA+FTS module will continue to execute until a point of convergence is reached.
- Convergence can be achieved when there is no significant change in the value of MSE over previous generations or by prefixing the number of iterative cycle.
- In our case although we have prefixed the number of cycles to 100 we observe that convergence is reached during 45<sup>th</sup> iteration only(around 338.61).

```
Generation : 85 = > 338.6084406314806
Generation : 86 = > 338.60841965705004
Generation : 87 = > 338.6083991648058
Generation : 88 = > 338.6083791383092
Generation : 89 = > 338.6083595618604
Generation : 90 = > 338.6083404204574
Generation : 91 = > 338.60832169975754
Generation : 92 = > 338.60830338604177
Generation : 93 = > 338.60828546618126
Generation : 94 = > 338.6082679276057
Generation : 95 = > 338.60825075827427
Generation : 96 = > 338.6082339466477
Generation : 97 = > 338.60821748166256
Generation : 98 = > 338.60820135270717
Generation : 99 = > 338.60818554959855
Generation : 100 = > 338.6081700625609
```

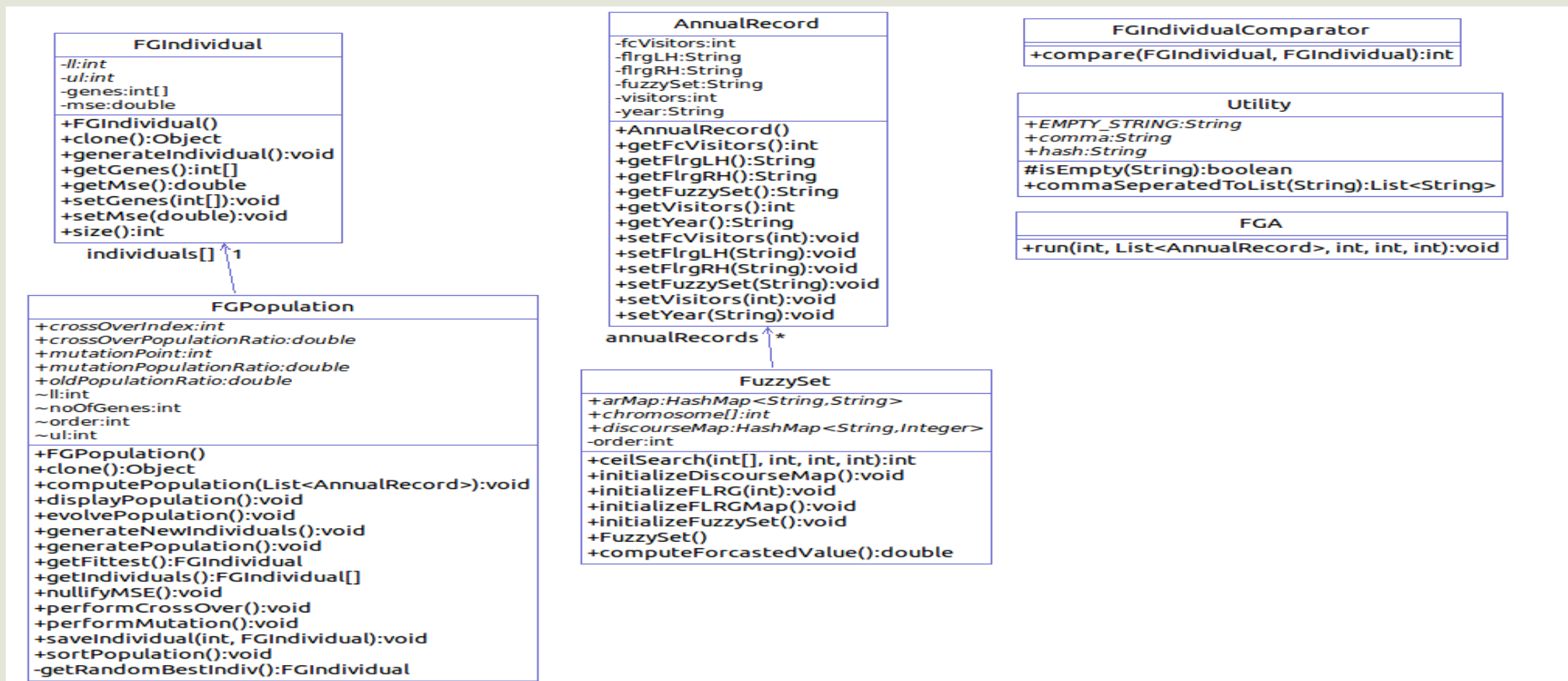
# Implementation : Result Visualization.

- Exploring different types of tools(Web Based and Desktop Applications) for result visualization.
- Chart.js : Makes use of HTML 5 canvas element.
- Play Framework: Lightweight , uses scala internally but Java interoperable.
- PlotLy: Web Based tool consisting of online style charts and spreadsheet.

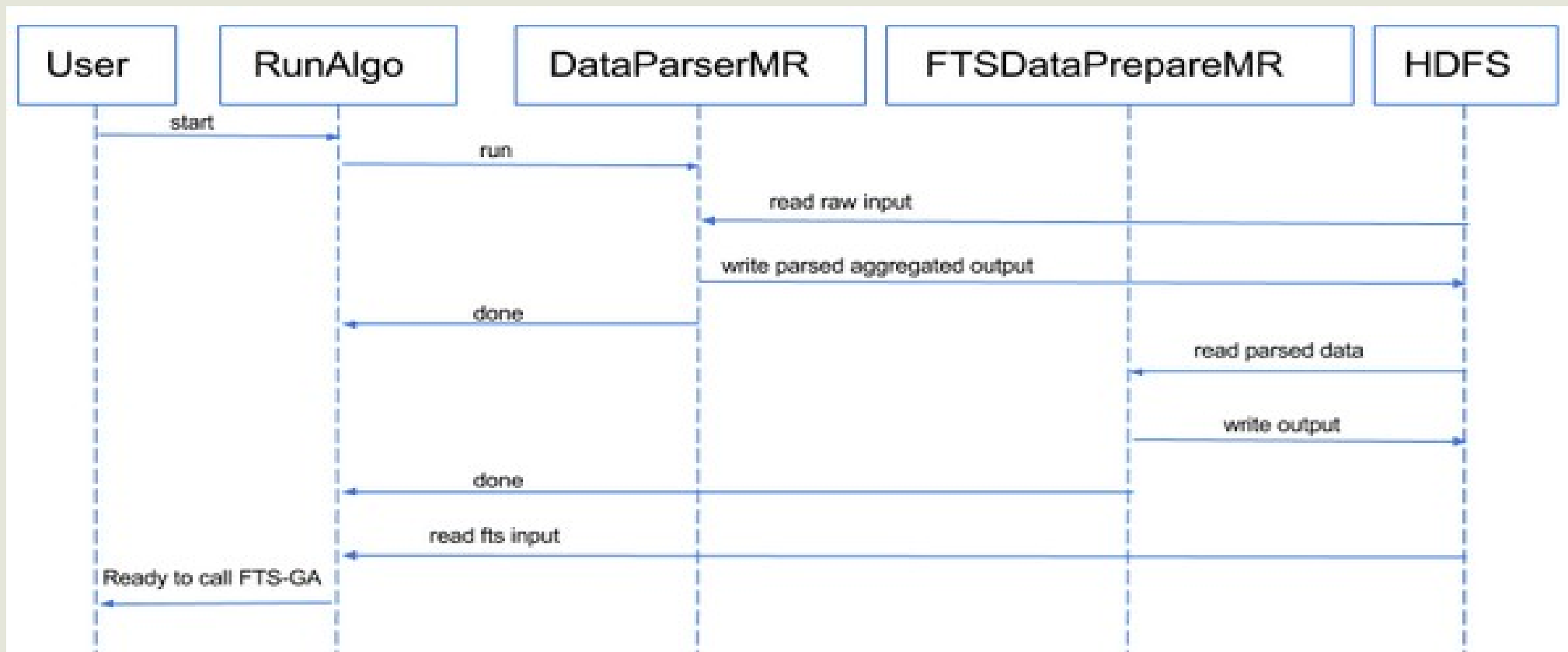




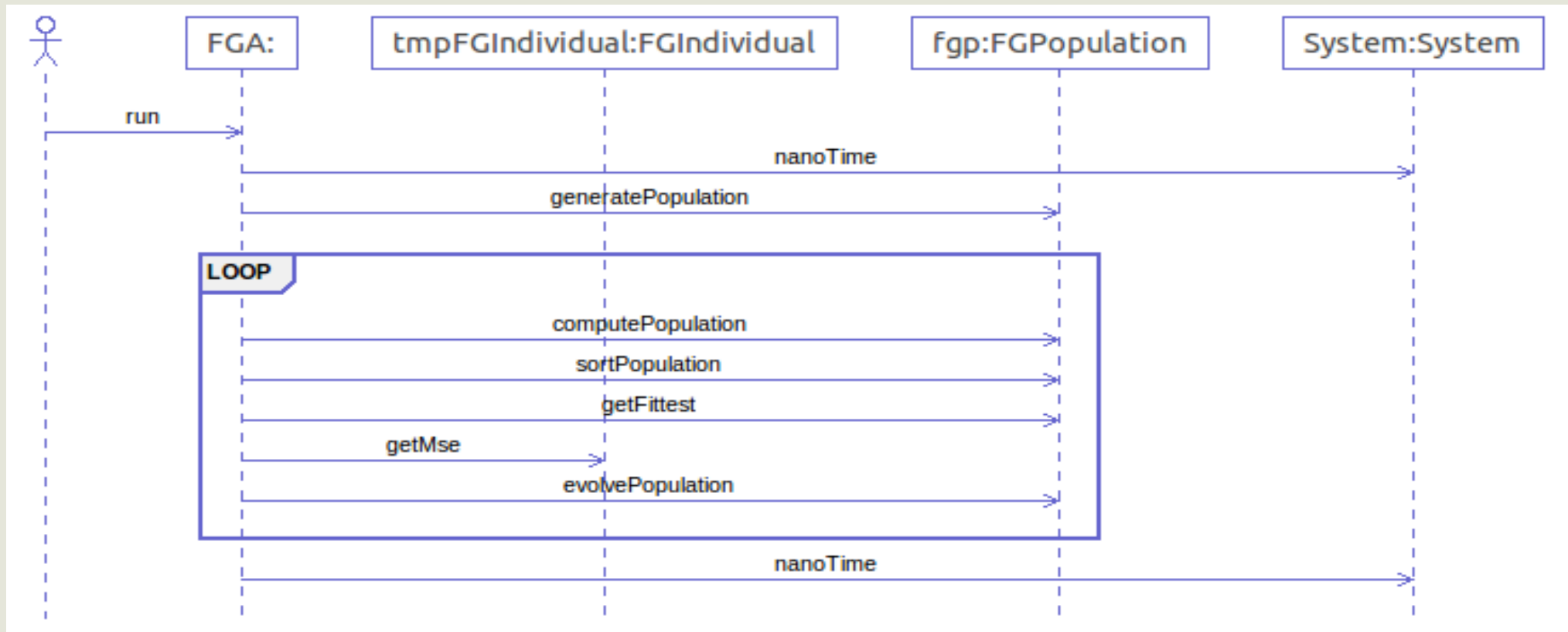
# Implementation: Class Diagrams



# Implementation : Interaction Diagram of Data Ingestion module



# Implementation : Interaction Diagram of Prediction Module



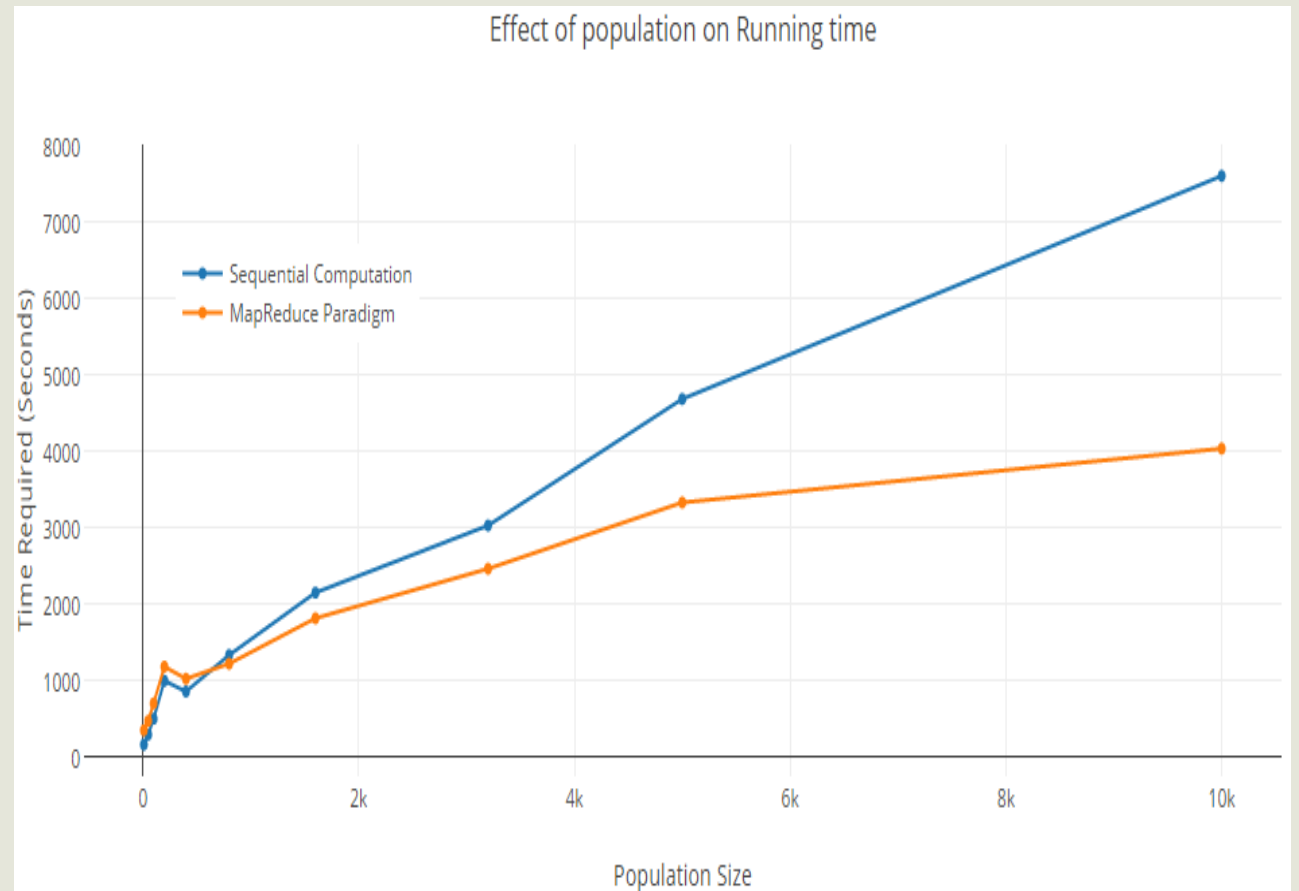


# Results

- Two main metrics for testing our forecasting models

1 . The accuracy would be evident by using historical data as training test data.

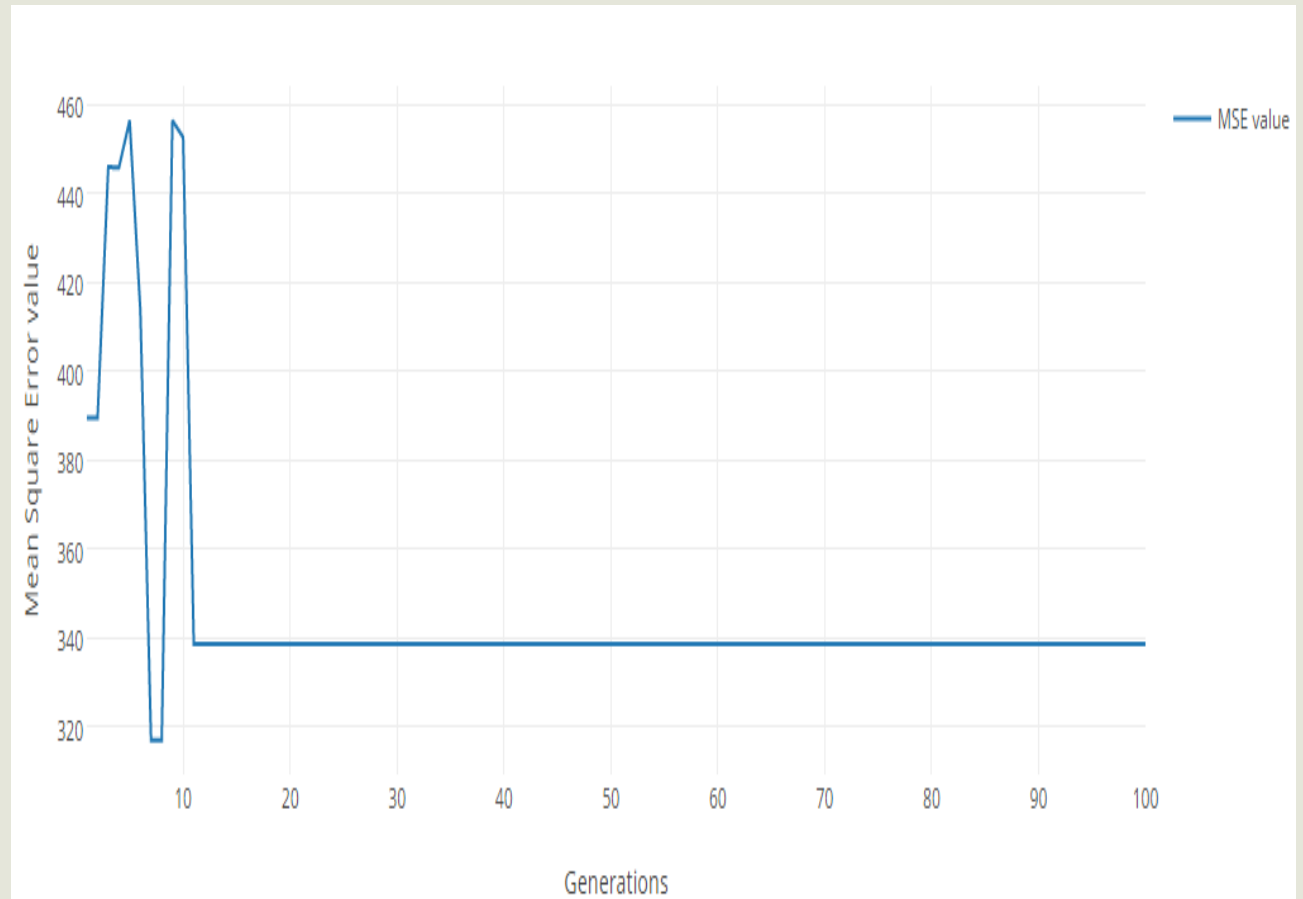
2. We intend to achieve is significant drop in running time by implementing the same in distributed/parallel.



# Observation

- Observed some major trends which are as follows:

1. As number of intervals in the universe of discourse are increased, the MSE tends to decrease.
2. As the order of fuzzy logical relationships increases, the MSE will decrease.



## Potential Future Scope of Work.

- Possible related work area is to use other parallel framework like Spark, Storm, Giraph or Hama.
- Initially only selection, crossover and mutation stages will be in reducer phase. We need to evaluate different mechanisms to improve the reduce phase.
- Currently we are manually tweaking these threshold values and visualizing the effects. We will try to have a dynamic way to sample training data and running multiple instance of the algorithm in parallel to decide appropriate threshold values.